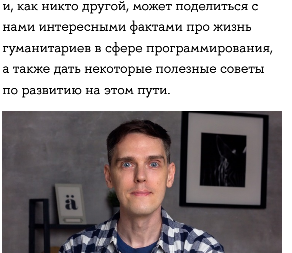
«Каждый человек может полюбить учебу и развитие, когда он поймет, как это делать правильно и эффективно»

Многие люди, приложившие руку к созданию «Фундаментальной и прикладной лингвистики», до сих пор остаются ключевыми фигурами в жизни студентов и преподавателей. Алексей Юрьевич Малафеев – один из них. На протяжении 5 лет он занимал пост академического руководителя программы, и, как никто другой, может поделиться с нами интересными фактами про жизнь гуманитариев в сфере программирования, а также дать некоторые полезные советы по развитию на этом пути.



Алексей Юрьевич Малафеев.

Я учился на филологическом факультете в лингвистическом университете, который закончил в 2006 году. Направление было исключительно гуманитарным. Там же я учился в аспирантуре, получил степень кандидата филологических наук. Однако программированием и компьютерными науками я интересовался с самого детства.

Первый компьютер у меня появился в 6 лет, тогда я впервые попробовал программировать, еще на языке Basic. У меня была советская книжка по программированию на этом языке, которую я изучал по мере сил. Мой компьютер был еще не на Windows; это был ZX-Spectrum с 64 килобайтами оперативной памяти. Учась в школе, я увлекался программированием, но не изучал его системно и целенаправленно, тогда меня гораздо больше интересовали компьютерные игры. Школьная программа по информатике давалась довольно легко, даже иногда решал олимпиадные задачи на Бейсике за школьников, но вот с математикой не ладилось, а в старших классах я пошел по гуманитарному профилю. В те годы я был убежден, что:

1) чтобы стать "настоящим" программистом, нужно быть с математикой на "ты";

2) математика - это не моё, у меня нет к ней способностей;

3) мне больше нравится заниматься языками, лучше стану переводчиком.

Следуя этим установкам, я больше развивался в гуманитарной сфере и поступил на иностранные языки. Я шёл на филологический факультет в иняз и был уверен, что работать буду переводчиком. На деле после окончания учебы я пошел сразу в преподавание. Сначала 4 года проработал преподавателем английского в языковой школе, немного преподавал японский в инязе, а потом меня пригласили работать в ВШЭ.

В ВШЭ я работал с 2010 года по 2020 год. Первые пять лет преподавал почти исключительно английский язык, потом меня сделали академическим руководителем образовательной программы «Фундаментальная и прикладная лингвистика». Я перестал преподавать английский и переключился на разные более или менее "компьютерные" дисциплины: введение в программирование для лингвистов, компьютерные инструменты лингвистического исследования, автоматическая обработка естественного языка.

Но как всё это получилось? Как преподаватель английского стал заниматься компьютерной лингвистикой?

Будучи студентом, я программировал исключительно для себя, это было просто хобби. Я любил писать несложные игры на разных языках, буквально что-то под руку подвернется, занимался этим не очень системно и серьёзно, в основном, "методом тыка". Много было "фейлов". Например, я приходил в книжный магазин и покупал какую-нибудь толстенную книгу по C#, дочитывал до середины, ничего не понимал, сдавался, откладывал, выбрасывал, а в итоге терял интерес на какое-то время. Часто думал: "Круто, наверное, быть программистом! Жаль, что я им никогда не буду..."

К началу моей карьеры в Вышке в моей жизни появился Python, я начал читать книжки по нему. Этот язык был более дружелюбен к новичкам, изучать его легче и приятнее, чем C/C++, C# или Java. Мне попались интересные книжки про Питон, которые смогли увлечь. А самое главное, в тот момент я наконец-то осознал, что надо не только читать книжки, но и пробовать что-то делать самому, сразу же применять знания на практике. Это как с иностранным языком: если просто читать учебник или словарь без практики речевых навыков, то ничего не получится.

Важное качественное изменение произошло, когда я начал писать на Питоне небольшие программы для работы. Одна из первых – простенький генератор словарных тестов для занятий. Лень – двигатель прогресса, мне хотелось автоматизировать одну из скучных задач по работе. Она была совсем простая, я думаю, что каждый студент ФиПЛа к концу первого курса способен такую программу написать, если занимался программированием нормально и не отлынивал. Но мне на тот момент эта программа далась непросто, я потратил много времени. Помню то чувство: "Ура, заработало!" Я стал её применять в работе, показывал своим коллегам, преподавателям иностранных языков.

Небольшой, но реальный успех мотивировал двигаться дальше. Я продолжал изучать Питон, стал писать более сложные программы опять же для работы. Например, генератор предложений на английском языке. Это уже более сложная задача: нужно было автоматически создавать грамматически правильные и более-менее осмысленные предложения на английском языке.

Мой самый серьезный и сложный проект того периода был Exercise Maker, генератор упражнений по английскому языку в различных форматах (например, как на ЕГЭ). Его я разрабатывал примерно с 2013 по 2015. Главная "фишка" в том, что на вход можно подавать любые тексты на английском языке, дальше программа сама автоматически оценивает сложность текста и генерирует около десяти разных видов упражнений.

По Exercise Maker я сделал несколько публикаций, в том числе в зарубежном журнале и на лучших российских конференциях по компьютерной лингвистике. Моей бесплатной программой пользовались преподаватели английского из разных уголков России и мира – я получал фидбек от преподавателей из Сибири, Испании, даже из Кембриджа. Успех дал еще больше мотивации; мне захотелось стать экспертом в программировании и разработке приложений на Питоне, а также "прокачаться" в машинном обучении и искусственном интеллекте.

В тот период я прошел много курсов на Курсере. Самый мой любимый – это, конечно, Machine Learning от сооснователя Курсеры Эндрю Ына. Автор курса очень интересный человек и классный преподаватель, он стал для меня одним из образцов для подражания. Этот курс до сих пор существует и является одним из самых популярных на Курсере. Кроме курсов, много читал научной литературы по компьютерной лингвистике, регулярно посещал конференции. Руководя многочисленными студенческими проектами, я все лучше узнавал эту научную область, понемногу становился в ней экспертом. Меня стали приглашать на различные конференции уже в качестве рецензента. Коллеги из Вышки, а иногда и других вузов, предлагали стать соавторами своих статей, участвовать в грантах.

Проработав столько лет в вузе, опубликовав более 50 статей, я стал постепенно уставать от "академической" жизни. Многие мои знакомые с факультета информатики совмещали преподавание в Вышке с работой в компаниях, я слышал их истории о работе вне вуза и со временем осознал, что хочу перейти в "индустрию". Но признаюсь, что мне долгое время не хватало смелости для этого шага, казалось, что моих знаний и опыта недостаточно для работы в крупной IT-компании.

Как-то раз в начале 2020 года один мой коллега, работающий на удаленке в зарубежной IT-компании и параллельно преподающий в Вышке, спросил у меня, как бы я решил одну задачу, которая возникла у него на работе. Я просто рассказал всё, что знал об этой задаче. Оказалось, что это было собеседование, которое я прошёл. Меня позвали работать в ту компанию, и так я стал тоже совмещать две работы: академическое руководство программой ФиПЛ в Вышке и бэкенд-разработку большого и довольно амбициозного программного продукта. Уже через три месяца работы в компании мне предложили повышение до Tech Lead, то есть технического руководителя проектной группы, в которой было несколько фронтенд- и бэкенд-разработчиков. Тогда я понял, что у меня неплохо получается и я хотел бы перейти на фул-тайм в индустрию. Это был целый новый мир: я писал код, руководил группой, проводил частые совещания с командой и с американским заказчиком. В конце 2019-2020 учебного года я решил уйти из Вышки и сосредоточиться на новом карьерном треке.

Прошёл год с того времени, как я закончил работать в вузе на полную ставку. Сейчас я лишь руковожу некоторыми курсовыми и дипломными работами, участвую в исследовательских проектах. За это время я успел освоить много новых навыков и технологий и даже ещё два раза сменить место работы, о чем расскажу далее. Но сначала отмечу, что в IT-сфере люди вообще чаще переходят из одной компании в другую. Для многих это более быстрый путь развития: гораздо легче получить повышение в должности и зарплате, переходя в новую компанию, чем если оставаться на том же месте. Тем не менее, я думаю, что когда тебе удается найти компанию, где чувствуешь себя действительно комфортно (дело ведь не только и не столько в зарплате), то нет никаких причин, чтобы не задержаться в этой компании на несколько лет.

Моя вторая IT-компания, тоже зарубежная, занималась внедрением технологий искусственного интеллекта в сферу здравоохранения. Там я проработал всего четыре месяца (а в первой компании – около полутора лет). Мне не очень подходила корпоративная культура, там было слишком много совещаний и переговоров, так что почти не оставалось времени на то, чтобы писать код. Удачно сложилось, что как раз в тот момент, когда я стал подумывать о смене места работы, на меня через сайт LinkedIn вышли хедхантеры из компании Behavox.

На тот момент я ничего не знал об этой компании, но почитав отзывы на GlassDoor, я понял, что это очень серьезная организация, амбициозная и успешная. Для меня одним из важных показателей стало то, что Behavox несколько раз входил в топовые мировые рейтинги как одна из ведущих компаний в сфере искусственного интеллекта, несмотря на то, что Behavox существует лишь около 7 лет. Я согласился на собеседование без особых надежд, но и риски были небольшие: у меня уже была работа, никакой срочности не было, я всегда мог найти что-то еще, это был лишь вопрос времени. Наверное, поэтому я чувствовал себя расслабленно и уверенно на собеседовании в Behavox.

Обычно трудоустройство в IT-компанию, особенно зарубежную, – процесс затяжной. Например, при устройстве в мою вторую компанию я прошел шесть собеседований и сделал тестовое задание; все вместе заняло чуть больше месяца. А в Behavox был так называемый "спринт" по найму: окончательное решение о найме кандидата принимается в течение недели. Я также прошел несколько непростых собеседований и сделал тестовое задание, но это происходило в сжатые сроки, что, на самом деле, менее стрессово, чем быть в "подвешенном" состоянии целый месяц. Так получилось, что я в итоге из сотен кандидатов со всего мира сделали очень щедрый "оффер" и я стал Senior Data Analytics Application Developer в Behavox. Сначала я взял себе небольшой отпуск, а сейчас уже работаю с новой командой, знакомлюсь с новыми проектами. Это очень приятный опыт, и меня особенно радует, что у меня получилось воплотить то, о чем я мечтал почти всю жизнь, но долго не решался. Это очень важный признак.

Я всегда с теплом вспоминаю ФиПЛ, и больше всего для меня всегда были важны именно студенты. Работая в Вышке, я стремился построить такую образовательную программу, на которой я бы сам с удовольствием учился, будь у меня такая возможность. Анна Юрьевна Хоменко, которая сменила меня на должности академического руководителя, на мой взгляд, действует в таком же ключе, и у нее это отлично получается. Мне кажется, что наша программа действительно классная. Она соответствует современному запросу на универсальных специалистов, лингвистов-программистов, во многих компаниях сейчас такие востребованы.

Лайфхаки для развития в сфере IT от ФиПЛовца

1. Система. У меня было много неструктурированного бессистемного самообразования, это неэффективно. Я даже считаю, что, как это ни печально, мое настоящее обучение началось только после окончания университета. Не повторяйте мою ошибку, учитесь сейчас и делайте это системно, осознанно, целенаправленно. Определите, что вам наиболее интересно, составьте реалистичный план развития именно в этом, и вперед. Также пробуйте (и не бойтесь) переставать использовать неэффективные и прибавлять новые (в дальнейшем), что в плане развития эффективнее для вас. Курсы? Тьюториалы? Ютуб? Книжки? Максимизируйте эффективные методы и подходы. Есть очень хороший курс про то, как учиться: "Learning how to learn" Барбары Оакли на Coursera. Круче методов из этого курса ничего не придумать.

2. Специализация. Информационные технологии – это океан, в котором легко утонуть. Я не знаю ни одного человека, который бы знал всё сразу в технологической сфере. Это не беда, нужно сосредоточиться на конкретной, более или менее узкой сфере и быть в ней экспертом. Лучше изучать одну тему – поглубже, чем немного по поверхности. Лучше хорошо разбираться в одном типе задач (например, автоматическая обработка текстов или распознавание звучащей речи), чем пытаться усидеть на нескольких стульях и заниматься сразу всем, от мобильной разработки до информационной безопасности.

3. Анализ рынка труда. Нужно завести профили на LinkedIn и hh, постоянно просматривать вакансии в этой сфере, подписаться на рассылки вакансий, освоить GlassDoor, – в комплексе это всё поможет. Есть даже отдельные курсы и книжки про то, как найти первую работу в IT, я читал много подобного, это крайне полезно.

4. Социальная сеть. Знакомства играют свою роль – не только в том смысле, что кто-то вас должен куда-то продвигать, нет. Общение с людьми, работающими в той же сфере, которая вам интересна, очень развивает и поможет во многих отношениях. Общайтесь с людьми из той сферы, где хотите работать, расширяйте сеть в LinkedIn, участвуйте в open-source проектах. К тому же, навыки общения нужно развивать специально, они востребованы везде, в том числе в работе программиста.

5. Портфолио. Нужно постараться уже за время учебы в вузе накопить некоторый опыт, собрать портфолио готовых проектов, а в идеале пройти стажировку или поработать парт-тайм в компании. Это не так сложно, как кажется, с опытом разработки собственных проектов придет уверенность, "прокачаются" навыки.

6. Установки. У вас вряд ли получится войти в IT-сферу, если вы туда идете исключительно из-за выгоды. С вероятностью 99% человек с неправильной установкой либо потерпит неудачу, не сумев преодолеть лень и сопротивление, либо, даже устроившись на работу, быстро выгорит.

Если вы действительно серьезно настроены работать в IT, нужно осознать и принять для себя две вещи. Первая: я добровольно соглашаюсь на то, что буду учиться всю свою жизнь, и никогда не будет такого периода, когда мне не надо будет учиться. Технологии очень быстро меняются, именно поэтому так важно постоянно развиваться в своей области. В связи с этим, вполне нормальная практика – постоянно проходить собеседования с целью поддержания формы. А в случае успешного прохождения можно даже улучшить свои условия работы.

Надо понимать, что если ты не любишь и не хочешь учиться и узнавать что-то новое, то, наверное, работа в IT – это не твое. Я считаю, что любой человек может полюбить учебу и развитие, когда он поймет, как это делать правильно и эффективно. Все рамки и страхи у нас в голове.

И второе, что нужно осознать: нам всегда кажется, что мы знаем будущее, мы знаем то, что с нами произойдет Б, если мы сделаем А, но на самом деле мы очень часто ошибаемся. Важно пробовать и не бояться, потому что ты никогда не знаешь, и никто не знает, что будет в итоге. Единственный способ узнать, что будет после А, – это сделать А.

7. Процесс, а не результат. Если ты все время делаешь то, что тебе не нравится, это тебя разрушает. Даже если ты убеждаешь себя, что ты приносишь пользу людям, но не получаешь при этом удовольствия от самого процесса, это долго без ущерба для тебя не продлится. Нужно очень осознанно настраиваться на процесс, а не результат. Это навык. Нужно научиться получать главное удовольствие и удовлетворение именно от того, что ты делаешь, а не от того, что бывает после (например, денежное вознаграждение или высокая оценка).

8. Soft skills. На собеседовании могут дать такую задачку, которую ты не можешь решить. Но это ещё не конец света. Ты терпишь неудачу на собеседовании именно тогда, когда нарушается твоя коммуникация с интервьюером. В целом, если ты закрываешься от людей, боишься и не идешь на контакт, ты не сможешь работать в команде и решать сложные, масштабные задачи. Поэтому, как ни странно, работа с компьютером требует социальных навыков и в целом soft skills. Эти гибкие навыки нужно целенаправленно развивать. Программирование будет занимать только 60% времени, в некоторых случаях даже меньше, потому что будет много совещаний, обсуждений, общения. Не стоит думать, что для работы нужны только технологии, только hard skills.

9. Победить синдром самозванца. Синдром самозванца очень распространен в IT. Это не только про неопытных людей, которые недавно выпустились из университета. Эта проблема может быть и у сотрудника с 10-летним стажем, которого только что повысили в должности и поручили новые, непривычные задачи. Это реальная общечеловеческая проблема, которая нас постоянно преследует и не даёт спокойно решать задачи и развиваться. Как с этим бороться? Синдром самозванца – это нормально. Чтобы справиться с ним, нужно вспомнить о том, что у тебя уже получалось, что ты уже знаешь и неплохо умеешь. Да, я не знаю и не умею А, Б и В, но зато у меня есть опыт решения вот таких задач, есть вот такие знания и навыки. Все, чего я не знаю, я могу изучить, как изучил уже что-то другое раньше. У всех нас есть успешный опыт изучения чего-то нового и сложного, нужно лишь напомнить себе, что когда-то и то, что я сейчас хорошо умею, мне казалось сложным и даже невозможным. Главное – не страдать, а предпринимать конкретные шаги для решения проблемы. Сама она не пройдет, а лишь усилится, поэтому надо с этим работать.

10. Приручить страх. Ничего не надо сильно бояться. Да, страх – это тоже нормально. Нужно относиться к нему проще. Как правило, подготовка и какие-то конкретные шаги для решения проблемы помогают справляться со страхом. Например, студент может бояться какого-то экзамена или защиты диплома. На работе вам также могут дать пугающе сложную задачу, но которую, тем не менее, необходимо выполнить. И как с этим быть? Все просто – надо разделить задачу на маленькие шаги, мини-задачи, и, как говорится, есть слона по кусочкам.

Любое большое дело можно разбить на маленькие шаги и сделать. Написать ВКР – это значит открыть текстовый редактор, создать новый документ, оформить титульный лист по шаблону (некоторые оставляют это на потом, но можно сделать сразу и получить небольшой мотивационный бонус, ведь успешное решение даже мелких задач всегда повышает настроение и прибавляет сил), набросать примерное содержание, написать первый абзац в разделе, который кажется самым легким для вас, и так далее. Начать изучать что-то с нуля – аналогично: 1) находим ресурс для изучения, 2) читаем первую часть/главу, смотрим первые пять минут курса/видео, 3) делаем первое упражнение, 4) (продолжите сами)... Полезно написать для себя список подобных мини-задач и отмечать, что уже сделал, это будет мотивировать.

В заключение хочу пожелать удачи всем, кто осилит этот длинный текст. Главное – это постоянно идти вперед, медленно, но верно, и все получится. Все обязательно будет хорошо!

Благодарим Алексея Юрьевича за такое увлекательное интервью и надеемся, что материал был полезен для прочтения.

С любовью,
Ваш ЖурPeople

Авторы: Мария Стягова, Александра Литвак 20ФПЛ

Академический руководитель
Климова Маргарита Андреевна

Менеджер
Жукова Валерия Валерьевна

Учебный офис

603155 Нижний Новгород,
ул. Львовская, 1В
каб. 145
тел.: 8 (831) 436-74-09 вн. 6535
vzhukova@hse.ru

Выразительная кнопка
для срочных сообщений

Программа «Фундаментальная и прикладная лингвистика» ВКонтакте
Нижегородская Вышка ВКонтакте
Нижегородская Вышка в Телеграм
Нижегородская Вышка в YouTube

Нашли опечатку?
Выделите её, нажмите Ctrl+Enter и отправьте нам уведомление. Спасибо за участие!
Сервис предназначен только для отправки сообщений об орфографических и пунктуационных ошибках.

Главная → НИУ ВШЭ в Нижнем Новгороде → Образовательные программы бакалавриата → Факультет гуманитарных наук (Нижний Новгород) → Образовательная программа «Фундаментальная и прикладная лингвистика» → Новости → «Каждый человек может полюбить учебу и развитие, когда он поймет, как это делать правильно и эффективно»

© НИУ ВШЭ 1993–2024 Адреса и контакты Условия использования материалов Политика конфиденциальности Карта
Шрифты HSE Sans и HSE Slab разработаны в Школе дизайна НИУ ВШЭ